Application of AIoT systems in medicine with emphasis to Wearable HealthCare (WHC)

Radovan Stojanović
Faculty of Electrical Engineering & MECOnet
University of Montenegro
Podgorica, Montenegro
stox@ucg.ac.me

Jovan Djurković
MECOnet

Innovation Fund of Montenegro

Co-financed by the Innovation Fund of Montenegro

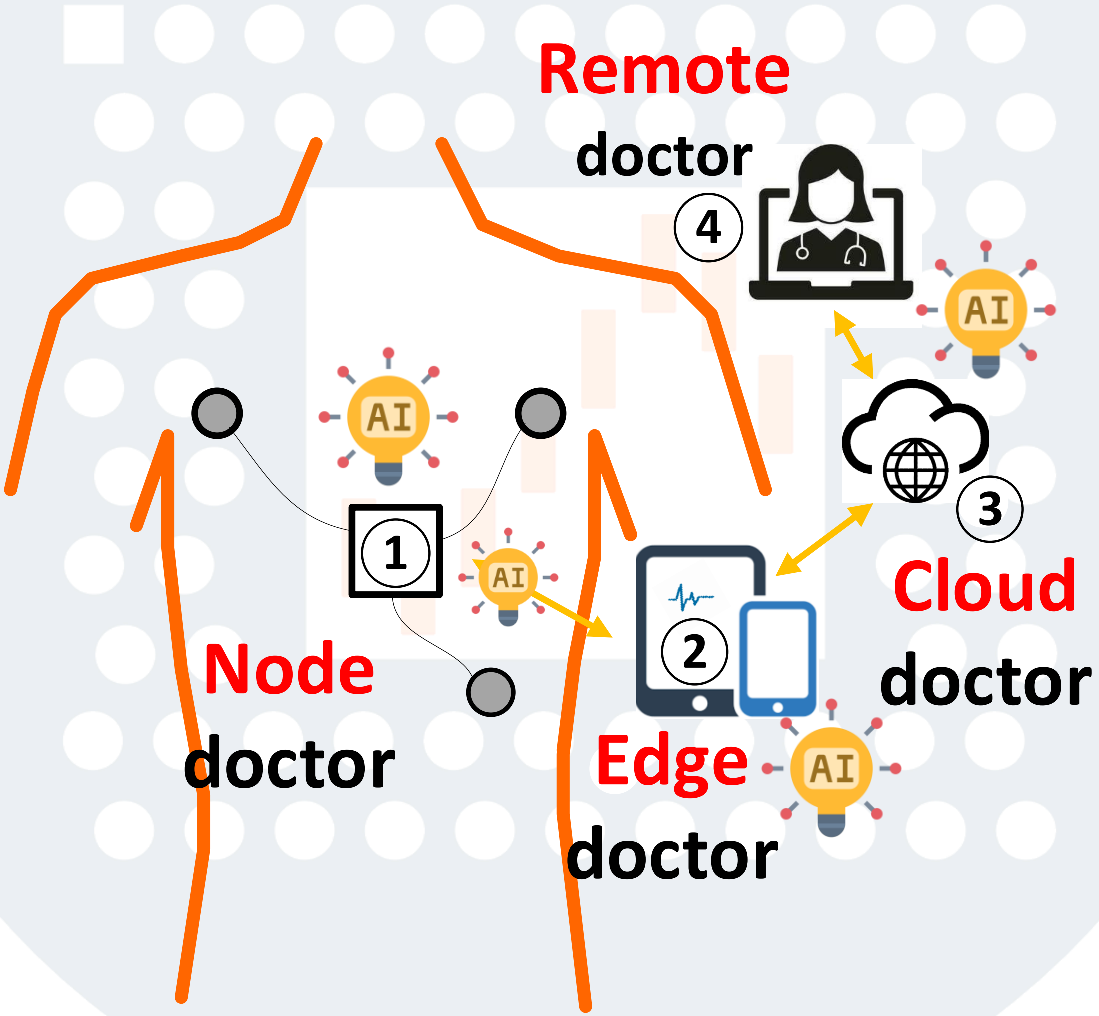
mant

AIoT EDU
EDUCATION ON AI & IOT
MONTENEGRO & REGION

Content

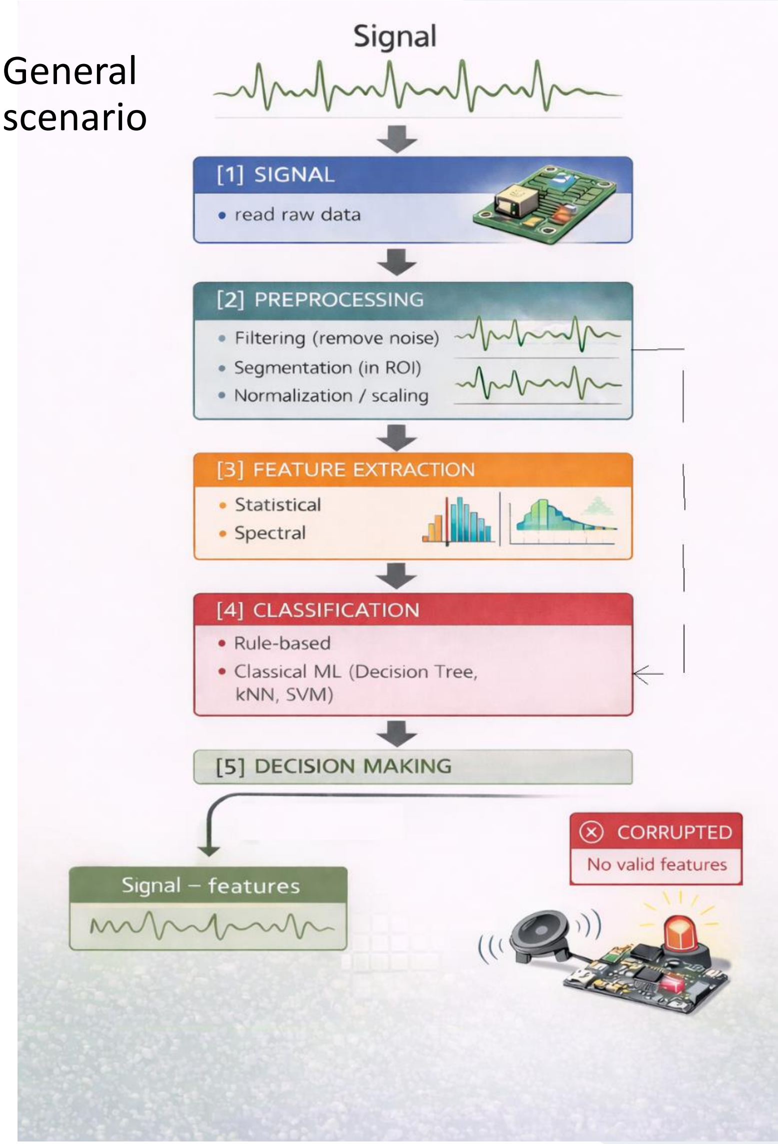
- AIoT in WHC
- AI pipelines
- Demonstration
- Examples
- Conclusion

AIoT in WHC


Remote
doctor
4
AI
AI
3
1
AI
Cloud
doctor
2
Node
doctor
Edge
AI
doctor

AI pipeline

- General scenario



- PPG signal scenario, fs=32Hz, 4s, 128 samples

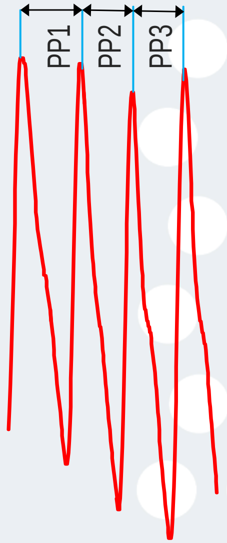
PPG Signal
|
▼
[1] Signal Acquisition
- PPG sensor reads raw data
|
▼
[2] Preprocessing
- Filtering (remove noise)
- Segmentation (e.g., 4s windows)
- Normalization / scaling
|
▼
[3] Feature Extraction
- Mean, SD, RMS
- Kurtosis, Entropy
|
▼
[4] Classification
- Rule-based (if SD > threshold → corrupted)
- Classical ML (Decision Tree, kNN, SVM)
|
▼
[5] Decision Making
- Clean / corrupted signal
- Trigger system action

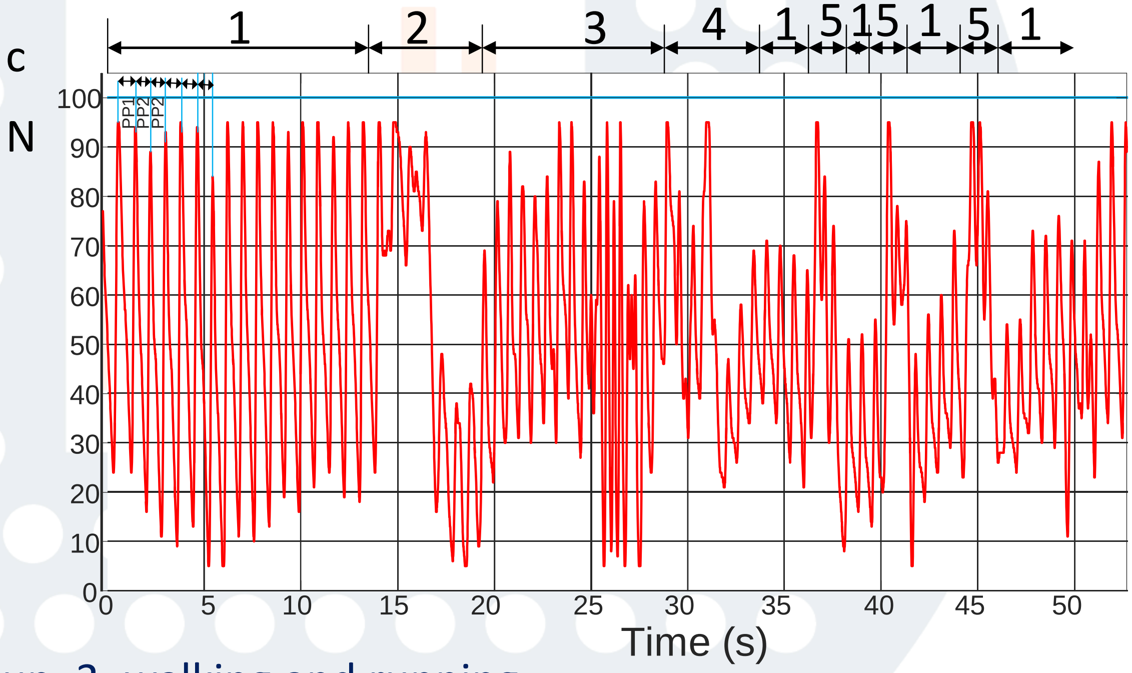
AI pipeline

- Historical perspective (BME)
- 1950s–1960s: Signal Acquisition & Filtering
  - First ECG, EEG, and PPG devices
  - Analog filters to remove noise
- 1970s–1980s: Segmentation & Feature Extraction
  - Compute statistical features (mean, SD, RMS, kurtosis)
  - Early digital signal processing applied
- 1960s–1970s: Rule-Based Classification
  - Threshold-based detection
  - Example: "heart rate too high/low → alert"
- 1980s–1990s: Classical Machine Learning (CML)
  - Decision Trees (ID3, 1986)
  - k-Nearest Neighbors (1967)
  - Support Vector Machines (1990s)
- 2010s–Present ———— Embedded Microcontroller Implementation
  - Low-power MCUs: ATtiny85, Arduino
  - Pipelines: segmentation → normalization → feature extraction → rule-based/CML
- 2015s–Present ———— Stronger Architectures & Edge AI
  - 32-bit MCUs: STM32, ESP32
  - Nordic nRF52/nRF53 (Bluetooth + sensors)
  - Grove Vision & AI boards (camera + sensor fusion)
  - Can run lightweight neural networks (TensorFlow Lite, Edge Impulse)
  - More complex pipelines: feature extraction + small neural networks for classification

AI pipeline

- **Practical problem**
  - 1D PPG signal N(t), corrupted by different artifacts classified in the classes c=[1,2,3,4,5] and summarized in two classes [1,2], 1 (Clean), 2 (Corrupted). It is important to know if signal is clean that we can accurately calculate PP1, PP2, PP3…and further features (HR, STDNN, RMSSD etc). Can we by AI determine classes 1,2,3,4,5 and finally 1.2 for signal N(t)?



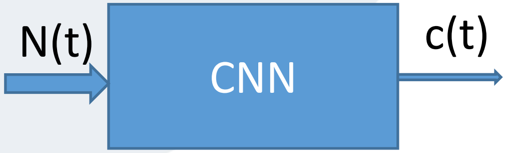


1=Normal signal, 2=Standing up, 3=walking and running

4=Seating down, 5=Cough ->

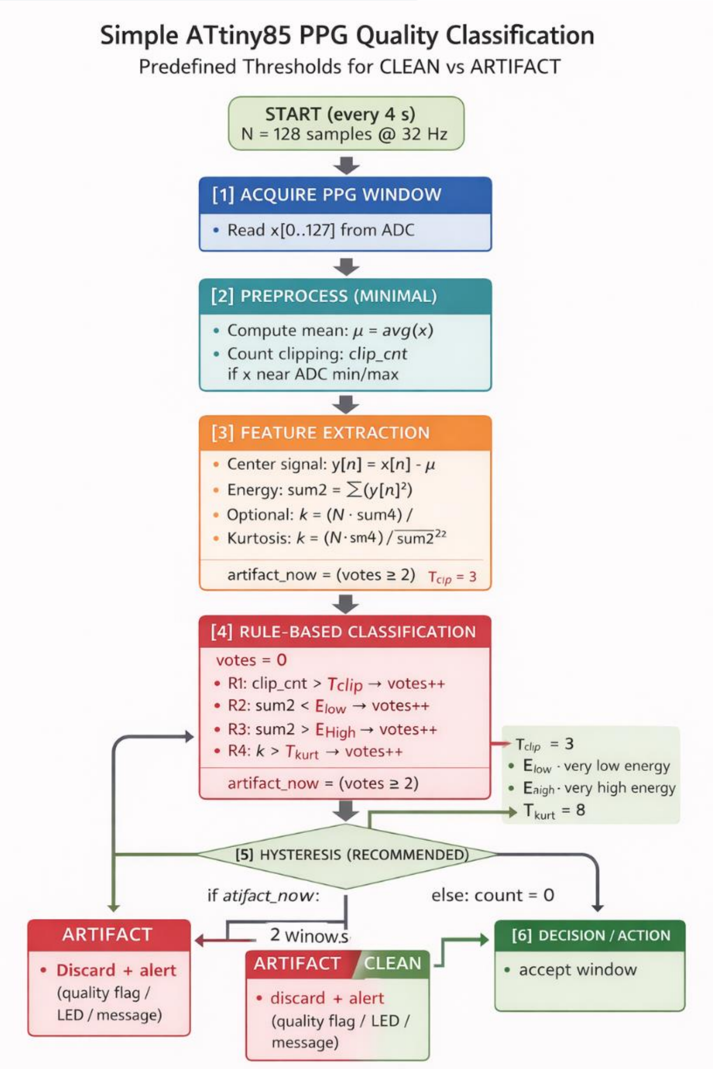
1=Clean, 2=Corrupted

Recorded by CardiaFylax, MECOnet, 100Hz



AI pipeline

- Scenario: Rule based
- App.: WHC
- Processor: ATTiny85, 20 MHz clock, 8 KB flash, and 512 B SRAM.Has a 10-bit ADC (good for reading PPG), a few I/O pins, and low power modes for battery operation.No floating-point hardware, so use integer math for calculations.
- Rule-based signal processing, small feature extraction, and simple thresholds, limited for neural networks or complex ML.


Simple ATtiny85 PPG Quality Classification
Predefined Thresholds for CLEAN vs ARTIFACT
START (every 4 s)
N = 128 samples @ 32 Hz
[1] ACQUIRE PPG WINDOW
• Read x[0..127] from ADC
[2] PREPROCESS (MINIMAL)
• Compute mean: $\mu = avg(x)$
• Count clipping: *clip_cnt* if x near ADC min/max
[3] FEATURE EXTRACTION
• Center signal: $y[n] = x[n] - \mu$
• Energy: sum2 = $\sum(y[n]^2)$
• Optional: $k = (N \cdot sum4) /$
• Kurtosis: $k = (N \cdot sm4) / \overline{sum2}^{22}$
artifact_now = (votes ≥ 2) $T_{clip} = 3$
[4] RULE-BASED CLASSIFICATION
votes = 0
• R1: clip_cnt > $Tclip$ → votes++
• R2: sum2 < E_{low} → votes++
• R3: sum2 > E_{High} → votes++
• R4: k > T_{kurt} → votes++
artifact_now = (votes ≥ 2)
T_{clip} = 3
• E_{low} · very low energy
• E_{aigh} · very high energy
T_{kurt} = 8
[5] HYSTERESIS (RECOMMENDED)
if *atifact_now*:
else: count = 0
2 winow.s
ARTIFACT
• Discard + alert (quality flag / LED / message)
ARTIFACT CLEAN
• discard + alert (quality flag / LED / message)
[6] DECISION / ACTION
• accept window

AI pipeline

- Scenario: Machine Learning (SML) with Convolution Neural Networks (CNN)
- App.: WHC
- Processor: ATTiny85.
- **(3) and (4), off-line PC or Cloud, handle datasets, running Python ML stacks, using a GPU and exporting a tiny model (TFLite/ONNX) or just weights.**
- **Best options**

1. PC / Laptop (local)Windows / Linux / macOSTools: Python + PyTorch or TensorFlow/Keras. Full control, data stays local, repeatable experiments.

2. Google Colab , Cloud, Easy, Free/paid GPUs available. Quick prototyping, sharing notebooks, training small CNNs easily.

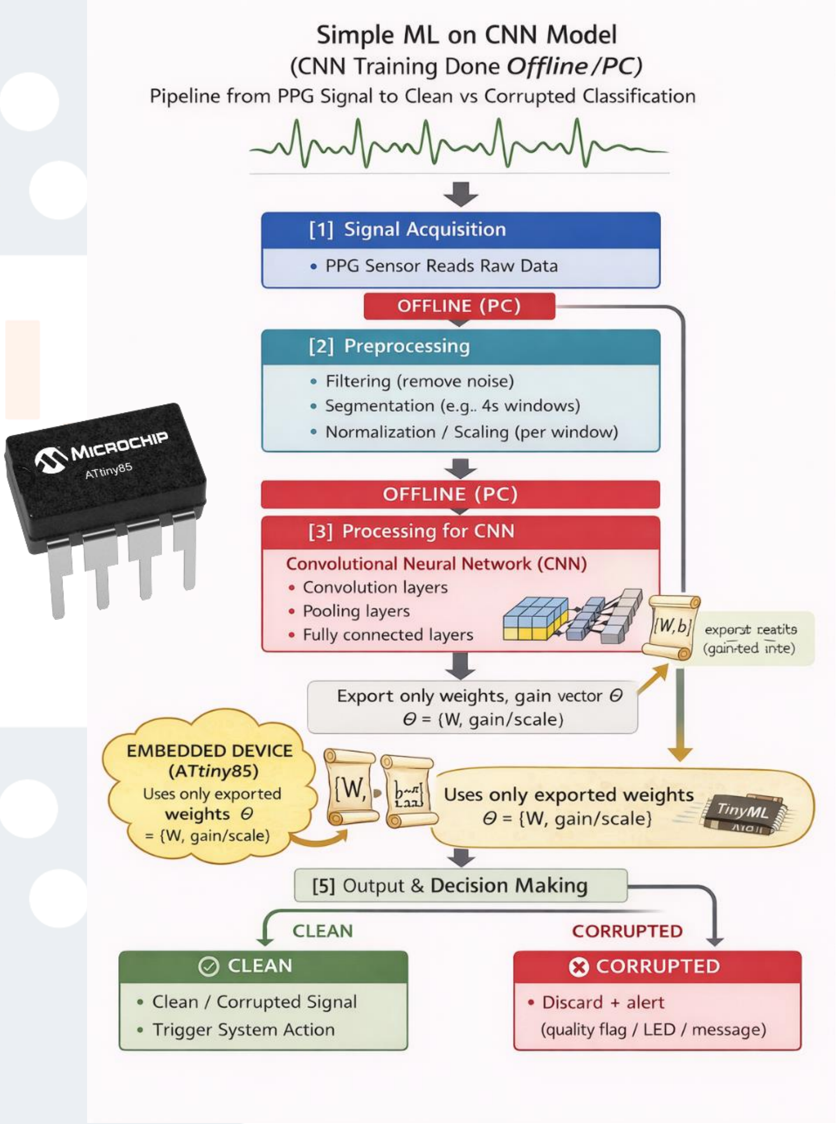
3. Kaggle Notebooks, Cloud, Eeasy)Similar to Colab. Good for running training jobs and keeping datasets organized. “Embedded-friendly” deployment tools (TinyML workflow)

4. Edge Impulse. very popular for TinyML). End-to-end: dataset. DSP/features. Model training. Deployment package, Great for fast pipeline building and exporting to embedded targets. Often used for sensor signals (time-series) like PPG.

5. TensorFlow Lite + TFLite Micro toolchain Train in TensorFlow/Keras (PC/Colab), export to TFLiteIf MCU is capable (not ATtiny85 usually), you can run inference via TFLite MicroBest for: Cortex-M MCUs (STM32, nRF52, ESP32, etc.). Good in case of prefering “classical ML then convert to rules” that is very good when target ATtiny85.

6. Scikit-learn (PC/Colab)Train Decision Tree / Logistic Regression then export: Decision rules (if-else)weight vector + bias (linear model)This is often the best bridge from ML to tiny 8-bit devices.

7. Matlab ML and NN toolboxes



Simple ML on CNN Model
(CNN Training Done *Offline /PC)*
Pipeline from PPG Signal to Clean vs Corrupted Classification

[1] Signal Acquisition
- PPG Sensor Reads Raw Data

OFFLINE (PC)

[2] Preprocessing
- Filtering (remove noise)
- Segmentation (e.g.. 4s windows)
- Normalization / Scaling (per window)

OFFLINE (PC)

[3] Processing for CNN
Convolutional Neural Network (CNN)
- Convolution layers
- Pooling layers
- Fully connected layers

Export only weights, gain vector Θ
Θ = {W, gain/scale)

EMBEDDED DEVICE
(*ATtiny85*)
Uses only exported
weights Θ
= {W, gain/scale)

Uses only exported weights
Θ = {W, gain/scale}

[5] Output & Decision Making

CLEAN

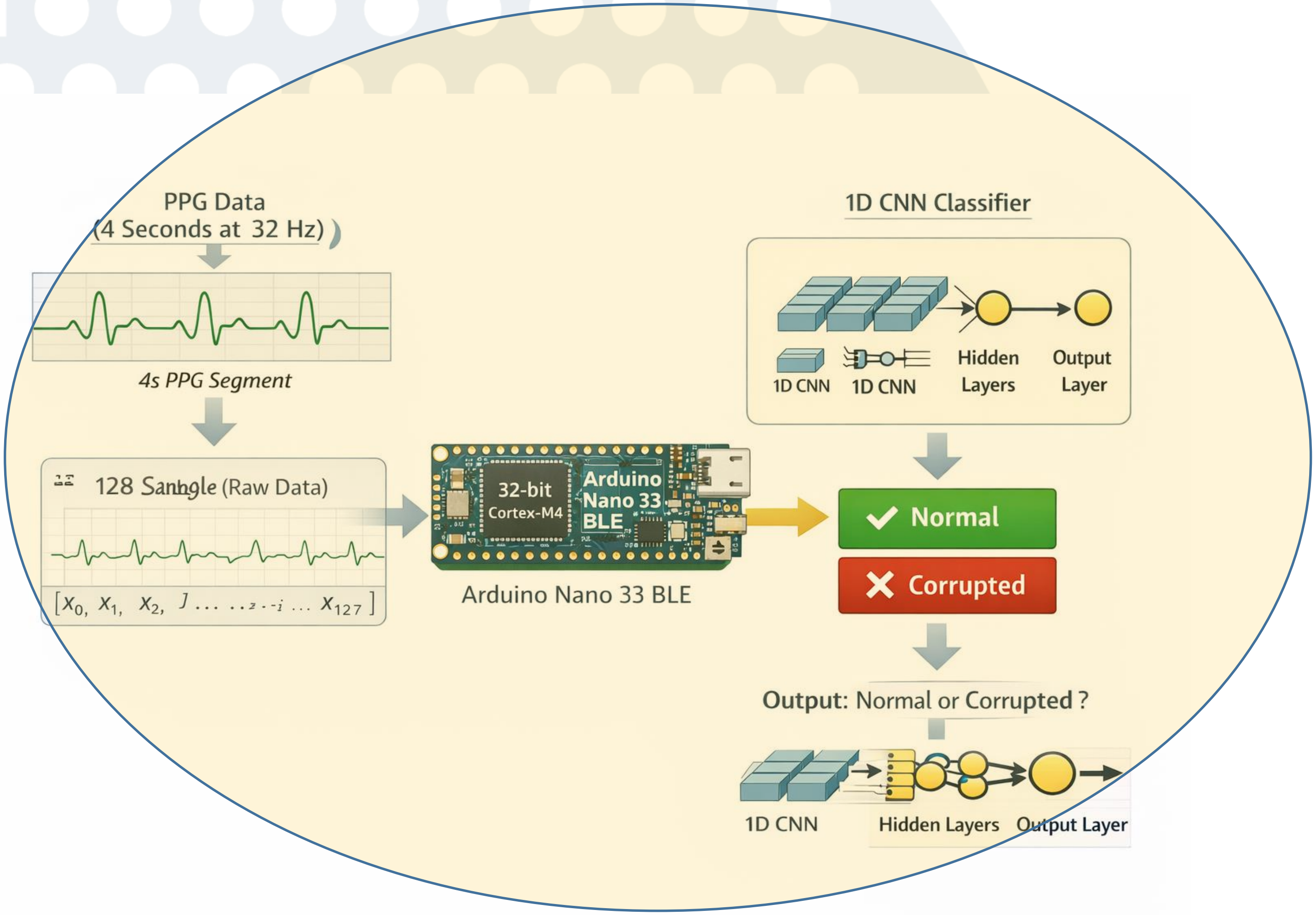
CORRUPTED

✓ CLEAN
- Clean / Corrupted Signal
- Trigger System Action

☒ CORRUPTED
- Discard + alert
(quality flag / LED / message)

AI pipeline

- Raw signal AI based CNN
- App.: WHC
- Processor: 32-bit ARM Cortex-M4 MCU (nRF52840 or nRF52833)
- 64 MHz clock.Has 256–512 KB flash and 64 KB SRAM, far more memory than ATtiny85. Includes Bluetooth Low Energy (BLE) support, multiple I/O pins, ADCs, and PWM channels. Can handle floating-point math and run TinyML or small neural networks on-device.Low-power modes make it suitable for wearables and battery-operated sensor systems.This makes it much better for real-time PPG processing with TinyML or sensor fusion compared to ATtiny85.



AI pipeline

Scenario: RAW Image
Processor: ATtiny + Grove Vision AI + MC

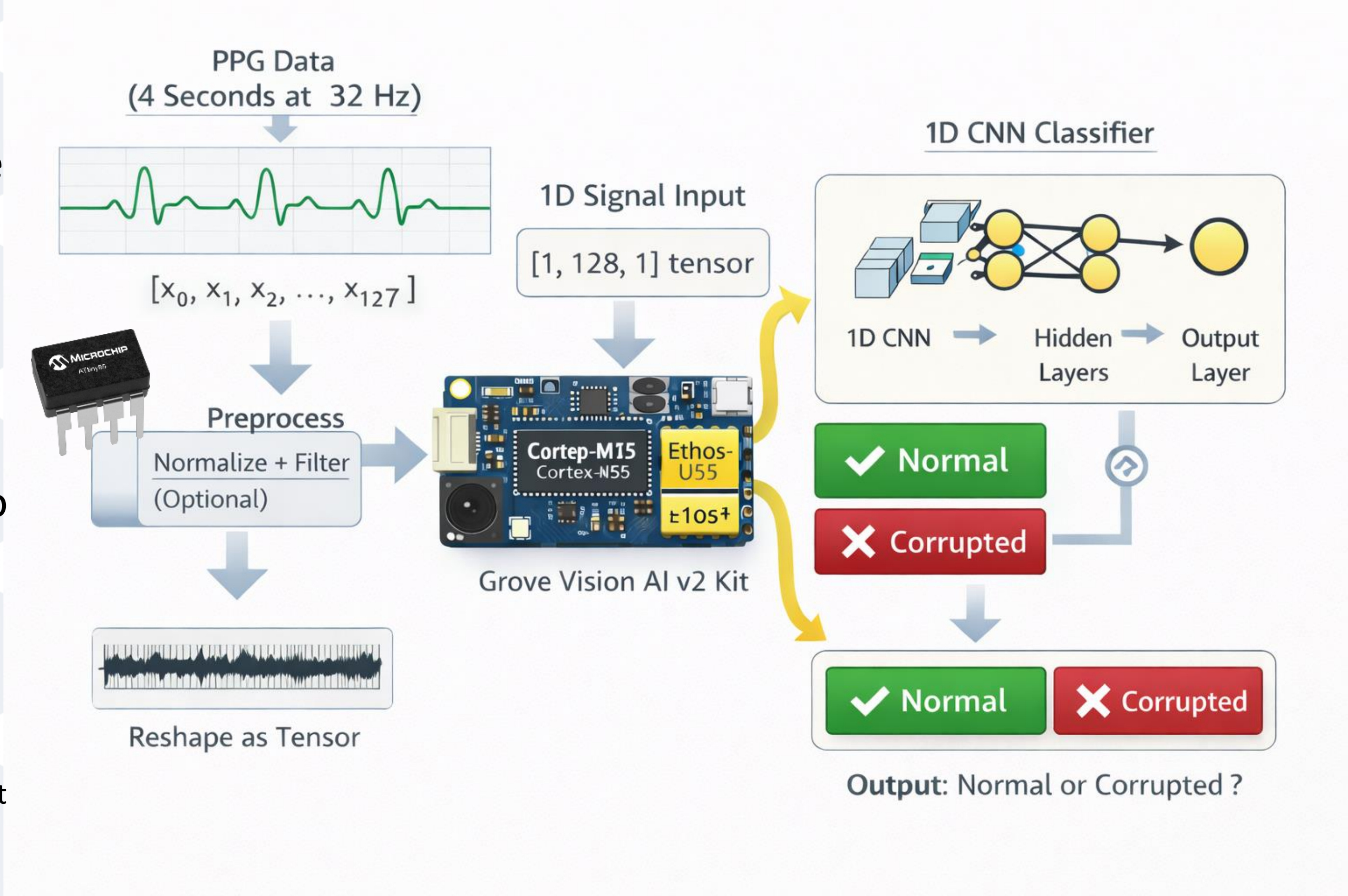
ATtiny85:

Sample 128 PPG points → store in array. Normalize / scale. Reshape as 1×128 tensor (array of bytes). Send to Grove Vision AI via UART/I2C

Grove Vision AI:

Receive 1×128 tensor. Feed into 1D CNN on **Cortex-M55 + Ethos-U NPU.** Output Normal / Corrupted

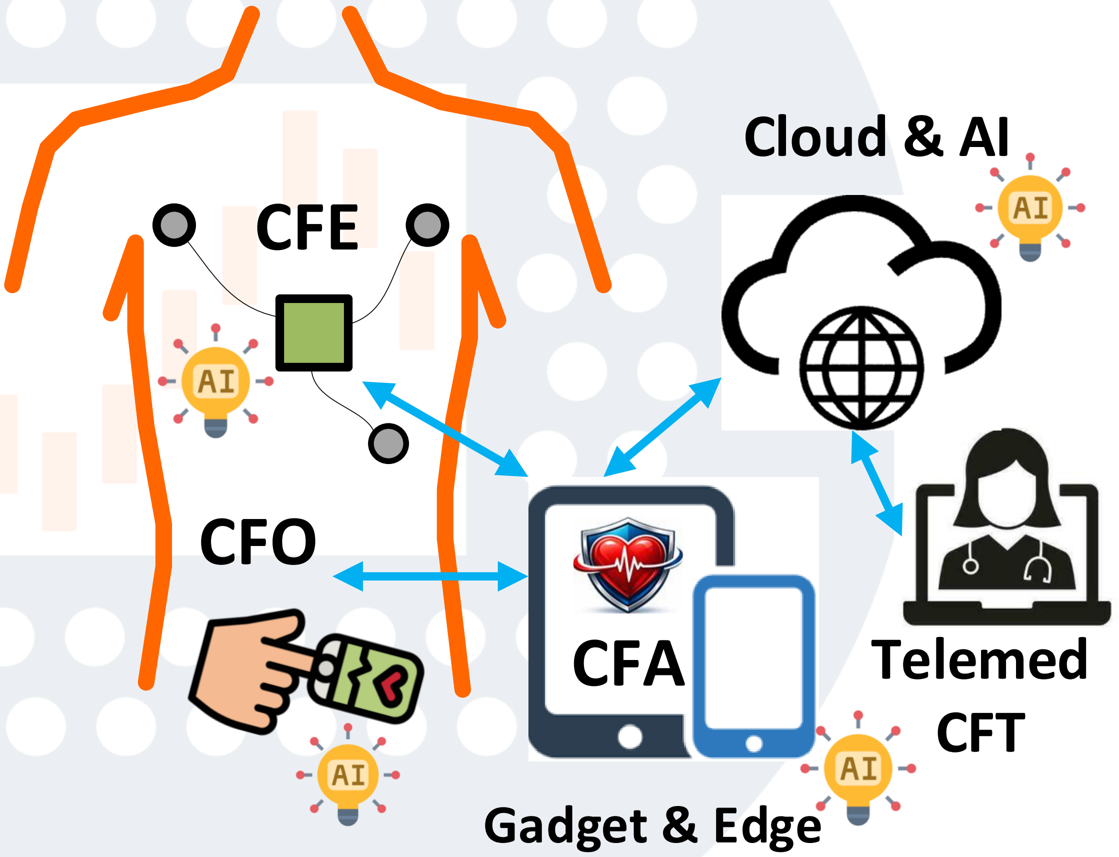
Grove Vision AI v2 Kit is an edge-AI module featuring a dual-core Cortex-M55 CPU and an Ethos-U55 NPU for fast, low-power inference. It supports 1D signals (like PPG or accelerometers) and small images, running TinyML models such as CNNs and dense networks. With 512 KB–1 MB RAM and 1–2 MB flash, it’s ideal for battery-powered AIoT applications. The kit integrates with Arduino IDE, Edge Impulse, and TensorFlow Lite Micro, making it suitable for real-time signal or vision-based classification and anomaly detection.



Demonstration

- CardiaFylax. Wearable AI and Telemed supported system for monitoring and analyze of cardiovascular and respiratory parameters

- Demo Video



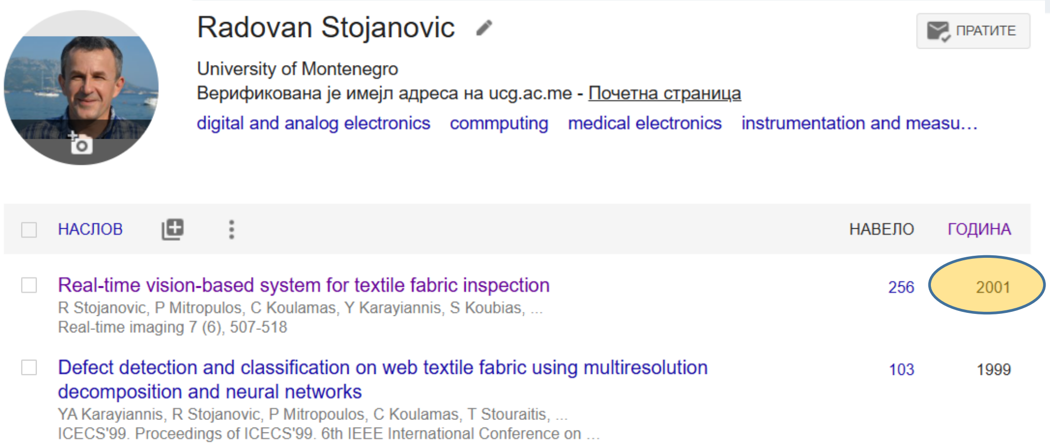
CFE – CardioFylax ECG Device, CFO – CardioFylax Oxigen Stauration Device, CFA- CardioFylax Gadget Application, CFT – CardioFylax Telemed Application

Practical Examples

- Jovan Djurkovic, Radovan Stojanovic, MANT, MECOnet, *Artifacts detection and classification for purposes of ECG signal acquisition and processing*
- Dubravka Šćekić, Azra Rastoder, Stanka Mišeljić, MSc students, ETF Podgorica, *AI on the Edge, alati i koncepti za AIoT sisteme*

History perspectives

The principles remain, but their implementation technology change



Radovan Stojanovic

ПРАТИТЕ

University of Montenegro
Верификована је имејл адреса на ucg.ac.me - Почетна страница

digital and analog electronics   commputing   medical electronics   instrumentation and measu…

НАСЛОВ	НАВЕЛО	ГОДИНА
Real-time vision-based system for textile fabric inspection R Stojanovic, P Mitropulos, C Koulamas, Y Karayiannis, S Koubias, ... Real-time imaging 7 (6), 507-518	256	2001
Defect detection and classification on web textile fabric using multiresolution decomposition and neural networks YA Karayiannis, R Stojanovic, P Mitropoulos, C Koulamas, T Stouraitis, ... ICECS'99. Proceedings of ICECS'99. 6th IEEE International Conference on ...	103	1999

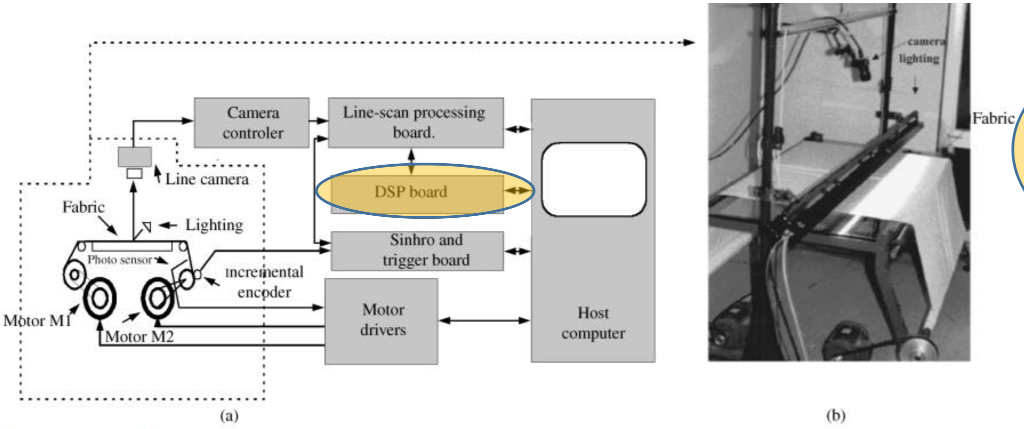


Figure 1. (a) Architecture of the implemented system; (b) unwinding machine.

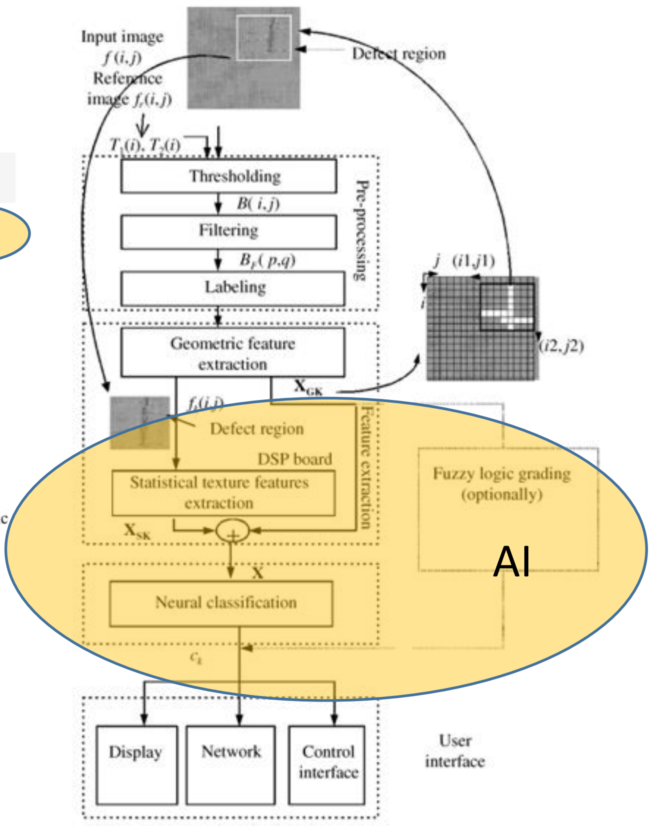


Figure 3. Flow diagram of the proposed inspection algorithm.

ACKNOWLEDGMENT

This Project

Edukacija o primjeni Vještačke inteligencije u Internetu stvari – AIoT Edu,

Education on the Application of Artificial Intelligence in the Internet of Things,

Contract #: S3EDU-034-25

is supported by

Innovation Fund of Montenegro

for whose support the authors are very grateful

Fond
za inovacije
Crne Gore

Thank you, Q&A?